Amendments to the CLAIMS:

Without prejudice, this listing of the claims replaces all prior versions and listings of the claims in the present application:

LISTING OF CLAIMS:

1. (Currently Amended) A method of reducing the amount of computer memory ~~utilised~~ utilized in calculating a formula on a collection of a plurality of series of data values, the method comprising the steps of:

(a) for each data value member of ~~a first~~ at least one series of said collection, determining the size of a window ~~around a current data value member~~ of data values required to calculate said formula;

~~(b) utilising~~ (b) obtaining the size of the window required for the at least one series on the basis of said determination, and

(c) utilizing said window having said predetermined size to determine data values to be stored in computer memory when calculating the formula when applied to other series of data values in said collection.

2. (Currently Amended) [[A]] The method as claimed in claim 1, wherein said step [[(b)]] (c) ~~comprises,~~ for members within a series of data values, ~~(a) for a first one of~~ in said ~~members~~ collection, includes:

(a) utilizing ~~utilising~~ said window to determine a first initial set of data values in the series to be stored in a portion of computer memory; and

(b) for subsequent ~~current members~~ sets of data values, loading the data values into the same portion of computer memory over locations ~~over~~ of previously loaded data values.

3. (Currently Amended) ~~In a~~ A computer system, with a primary memory store, comprising:

a ~~method of~~ computer program having a process for carrying out a formula calculation on a collection of series of data values, the calculation being carried out using members of the

series with the calculation for a current member of the series being <u>dependent</u> ~~dependant~~ on other data values located relative to the current member of the series, the ~~method comprising the steps of~~ <u>process including</u>:

(a) for ~~a given current~~ <u>each</u> member of said series, determining ~~from~~ <u>a window size corresponding</u> ~~the formula~~ <u>to</u> a relative series of consecutive data values required for determining said formula for said ~~current~~ member;

(b) ~~for each current member of said series of consecutive data values: (i) ensuring a corresponding relative series of consecutive~~ <u>obtaining the size of the window required for the series on the basis of said determinations; and</u>

(c) <u>utilizing said window having said predetermined size to determine</u> data values to ~~said current member are currently loaded into said first primary memory store; (ii) performing said formula calculation to determine a current output value~~ <u>be stored in computer memory when calculating the formula when applied to other series of data values in said collection.</u>

4. (Currently Amended) ~~A method~~ <u>The computer system</u> as claimed in claim 3, wherein said step [[(b)]] <u>(c)</u> ~~comprises~~ <u>includes</u>:

for a first one of said current members, loading into said first primary memory store ~~and~~ initial corresponding relative series of consecutive data values; and

for subsequent current members, loading only new members of said corresponding relative series into said primary memory store.

5. (Currently Amended) ~~A method~~ <u>The computer system</u> as claimed in claim 4, wherein said series of consecutive data values are arranged in a column.

6. (Canceled).

7. (Currently Amended) ~~A method~~ <u>The computer system</u> as claimed in claim [[6]] <u>3,</u> wherein new values are loaded into said primary store at an address determined by a modified modulo arithmetic operator which produces positive address values only.

8. (Currently Amended) <u>The method as claimed in claim 1, wherein the size of the window determined in step (b) is the minimum size required to calculate the formula for every data value in the at least one series</u> ~~A method of reducing the amount of computer memory required to be utilised in the calculation of a formula applied to a collection of series of data values, the method comprising the steps of:~~

~~(a) for a first member of said collection: determining a minimum window of data values required for storage in computer memory for the calculation of said formula when applied to said first series;~~

~~(b) for subsequent members of said collection: utilising the minimum window to determine data values of a series that need to be loaded into said computer memory for calculation of said formula.~~

9. (Currently Amended) [[A]] <u>The</u> method as claimed in claim [[8]] <u>1</u>, wherein said step [[(b)]] <u>(c)</u> further ~~comprises;~~ <u>includes</u> moving the series of data values through a fixed portion of computer memory, and <u>utilizing</u> ~~utilising~~ the data values ~~whilst~~ <u>while</u> in <u>the</u> computer memory to calculate said formula.

10. (Currently Amended) ~~A method as claimed in~~ <u>The computer system of</u> claim [[9]] <u>3</u>, wherein said step (a) further ~~comprises~~ <u>includes</u>:

calculating said formula on an initial data set to determine a minimum window size~~;~~ ~~the determination being~~ by ~~the steps of~~ providing an initial current window around a current data value to be calculated;

determining in accordance with the formula, other data values required,

<u>wherein</u>

if the other data values are below the current window<u>,</u> ~~then:~~ sliding the window down or making the sliding window bigger to encompass the other data values, ~~including~~ <u>and</u>

if the present sliding direction is to slide <u>the window</u> up, ~~then~~ making the window bigger to encompass the <u>other</u> data values, and if the window becomes large

enough to encompass the other data values, setting the sliding direction to be undefined[[;]], and

if the present direction is sliding up then sliding the window down by dropping previously calculated values from the top of the current window, provided that no value that is busy being calculated may be dropped off;

upon completion of the calculation of the formula, determining a final required window size;

subsequently, for each of the other data sets in the series, utilizing utilising the final required window size to calculate the formula for each data set[[,]] by moving through the data set element by element and storing values in the current window within the primary memory store.